

208



ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

May 1982
Mike Dunn & Jim Bumpas, Co-editors

23252



Too Much Adventuring-by Tom Ettel

News Bites

-by Mike Dunn, Co-Editor

Stacy Goff has resigned as president of ACE because of his need to travel alot; our new president is Kirt Stockwell, an instructor in Computer Science at our local Junior College. Kirt has written several articles on Structure in programming and Random Access, and will continue to do this. Marc Benioff now is programming for Automated Simulations, and will write for A.N.T.I.C. as well as ACE. Jerry White is writing articles for many publications. Stan Ockers will be found in A.N.T.I.C. and M.A.C.E. as well as ACE in the future. Speaking of M.A.C.E. (Michigan A.C.E.), they have taken a much different way than ACE, and have become a magazine with advertising, Bulk Mail (I just got the March issue!), and great articles. The March issue has a superb article 11 pages long on the new GTIA chip that is must reading for anyone interested. (M.A.C.E. POB 2785 Southfield, MI 48037 \$15 year, \$2 issue [March]).

In this issue, we have our first original hardware article, a Cassette interface board that will allow you to use any stereo cassette deck with your Atari. The schematic was sent to me by one of our Australian members, and I copied it the best I could. Recommended for advanced hobbiests until I find out more about how it works. Let me know how it works for you. More hardware articles would be very much appreciated by all.

This week Earl Rice and Marc Cator of the Atari User Support Group came to Eugene to speak to our local members and answer questions. It was an informative and pleasant evening for all; next they go to Portland and Seattle before heading back to Sunnyvale.

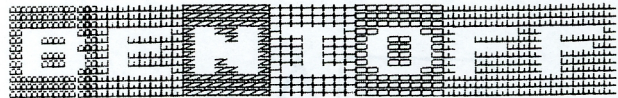
NEW BULLETIN BOARD !!

by Alan Ackezman
MICROBITS
434 W 1st
Albany, OR 97321
(503) 967-9075

I am pleased to announce a new bulletin system for Atari owners with modems. The Bulletin Board will offer many of the standard services (e.g. Messages, News) and some new ones. One of them, the Question and Answer service, will allow you to get help on your computer related problems such as programming, equipment and rumors. Our other new service, Ordering, will allow you to buy things from the comfort of your home. We will offer the best new products, reduced prices and specials as soon as they happen, and we have free delivery on all items. In the near future, program downloading and uploading will be implemented.

You can call the Microbits Bulletin Board at (503) 967-9075. The hours are from 6:30 pm to 9:00 am every day of the week. If the computer does not answer on the first ring the system is not operating for some reason. We would appreciate it if people using the system would send comments and suggestions for improvements and new features to SYSOP using the message function.

This is not the ACE Bulletin Board!



Benioff At Large - Mr. ATARI

By Marc Russell Benioff

Because the West Coast Computer Faire is just behind us, there is not too much ATARI news. So, I decided to review four of my favorite games that I picked up this month. The games that I will review are: 1) Threshold, 2) Millipedes, 3) Action Quest, 4) Trivia Trek. All of these games are superior, and really show what this machine is all about.

Threshold is by On-Line Systems. Ken Williams ("Mr. On-Line" as known among the ranks) told me at the show that, "Atari Threshold is much better than Apple Threshold. The main reason is because of the superior sound." I had played Apple Threshold many a time at my friend's house who has an Apple. It had already become my favorite Apple Game. When I booted it up on the ATARI, I nearly, not quite, fell off my chair. This game is EXACTLY like the Apple version, and the sound (done by John Harris) is great! The game itself is written by Warren Shraeder. Threshold begins as huge winged creatures fly above you as Galaxians, if you will. These are not normal Galaxians, but highly advanced, in patterns that will out do most arcaders the first time. These birds are only the first of 24 levels of creatures that attack in various patterns. After every four levels, your fuel is restocked, and the mother ship gives you a kiss. Does this sound awesome? Well, it is! But, there is a hitch!! You must have 48K, and one disk drive. But, besides that minor "flaw" it is my favorite game right now. Throw away Galactic Chase, throw away Space Invaders, THRESHOLD IS HERE!! As soon as I can get past the 16th level I will tell you, but it gets a bit hairy right around there!!

If you like ATARI Coin-Op Centipedes, YOU WILL LOVE MILLIPEDES!! For those of you who don't know what Centipedes is, it is little Centipedes coming down to get you. You must shoot these guys, and score points, while avoiding ticks, flees, and scorpions. This version is even better than the version to be released by ATARI in July. These bugs fine scroll across the screen, contrasting the course movement done by the soon-to-be released ATARI version on 8K cart. To achieve this fine scrolling, the author had to use one screen page for the mushrooms, and a second page for the centipedes, and unfortunately when he did this he created a slight flicker flipping between the screens, but besides this minor flaw, Millipedes is great. Ahh, but you ask, "WON'T ATARI SUE THEIR PANTS OFF?" YES!! That is why the product is being discontinued, and only 500 copies are out there!! Do not fret-On May 10, 1982 they will start shipping a new version called Train Robber, with the same routines, but different graphics!! Who are they? They are Software Street in Chino, CA. See addresse at end of article.

The third game I have recently received is ACTION QUEST!! I have to tell you, I love ACTION QUEST!! It goes on ANY ATARI machine with a cassette!! In Action Quest, you are on an adventure in a Haunted House. This just isn't any adventure, but, if you will, an arcade adventure!! You are a ghost trying to climb up five levels of rooms (6 rooms per level), and eventually tryink to escape. The things you have to accomplish in Action Quest are strange. In some rooms you have to shoot at things, while at others you must solve a mystery to unlock the prize in that room. Oh, yes, the object is to go into a room, touch a certain object, and leave without being killed. Sound easy? WRONG!! If you do not have an adventure mind with arcade reflexes forget it!! I have gone though the entire game, solved every mystery, and it took me a total of 3-5 hours! But, even though I have solved it, and know exactly what to do, I still go back and play it all the time!! Action Quest is a NEW game from a new company called JV software. Jack Verson wrote the game, and is currently working on a sequel.

Trivia Trek is by Jerry White, and is being sold through Swift Software. When I get tired of playing my arcade games and want something to stimulate my mind while still being entertained, I turn to Trivia Trek! TT is a quiz. You are given a category, or you can choose one, you must answer questions on that topic. The answers you choose are multiple choice, and response is controlled by the joystick, or key board!! There are mucho categories located on the 2 disk program, but when you are through with those (in about 42 years), you can design your own with a built in utility. Pretty neat, Huh! From the same company that brought you My First Alphabet, and from the man you have bought such games as! Name that Song, Trick Tutorials 6 and 7, and Sunday Golf, bring you TRIVIA TREK. I am sure the thinkers will enjoy Trivia Trek!

Other games I have recently recieved, and regard as top notch are! Apple Panic, Deluxe Invaders from Roklan, and Nominoes Jigsaw Puzzle from Artworx.

Recently I wrote a review of a game called, "Ali Baba" from Quality Software. I basically said it was a game with great sound, and graphics but to play you needed, "NERVES OF STEEL!!" Bob Christenson, owner of QS, approached me at the SF Faire, and said he didn't like my review. Well, I told him, "I didn't like your game, I paid \$45, and I AM ENTITLED TO MY OPINION!" He said that he agreed with me, about having the right, but still disagreed. I told him I would put this paragraph in my column, and would remind people that reviews I do are my opinion, and not anyone elses. I still don't like the game, and still is \$45 sitting in my disk box, never to be used again. Maybe I'll reformat it, then I will have lost only \$40, of course that is my opinion.

Feel free to quote me on any of my reviews. And I will have romors and facts next month in Benioff At Large-Mr.ATARI.

MARC RUSSELL BENIOFF

REVIEWS

DISKSCAN Author: DAVID YOUNG

If you ever wanted a powerful utility for examining and/or modifying sectors or files on a disk, this is it! I was first impressed with the documentation. It is on the disk; you print a full 16 pages of very readable user guide and an excellent tutorial about Atari disk data structures.

The program itself is menu driven, and very user friendly. There are default values for all requested input, and you may specify sectors in decimal or hex. You can display directory information (names, starting sectors, status, etc.), or display the contents or any sector (hex or characters at your choice). Changing data in a sector only requires positioning the cursor (screen editor), changing the byte(s) on the screen, and giving a write command. The most advanced feature of DISKSCAN, though, is the ability to do an on-screen disassembly of any machine language portion of the disk. You display a sector, tell the disassembler where to start, and watch the assembly language appear on screen beside the sector data display! Not only that, but DISKSCAN has an in-line assembler so that you can input assembly language to add or modify machine language at any place in any sector.

Another handy feature is the ability to dump any display to the printer. There is not space here to do more than mention some of the other features, such as search part or all of disk for a 1 or 2 byte sequence; transform section of code to a load file; image segments to another disk; etc. I believe this to be the most powerful tool yet for anyone who wants to get on really intimate terms with his disk based software.

DISKSCAN is marketed at \$45, but David is making it available to club members for \$25 postpaid, when ordered through him, at 421 Hanbee, Richardson, TX, 75080. Make checks payable to CDY Consulting.

Requires 32K (24K if you don't use the optional assembler/disassembler features).

--Bill Halberstadt--
505 Lark Drive
Newark, DE 19713

Letter Perfect-ROM version #2005 \$250

L.J.K. Enterprises
P.O. Box 10827
St. Louis, MO 63219-0827
(314)846-6124

WOW! Imagine plugging in a neatly packaged ROM, having your DOS and program booted instantly! The new Letter Perfect ROM does that and more; it is the first ROM based business program for the Atari, and the first word-processor that can be configured for any parallel printer. After turning on your computer, you are asked if you want one of the default printers (Atari/Centronics 737 & 739, Epson, Qume/Diablo, or Diskette. If you choose a default printer, you instantly get the LP menu and proceed. If you chose the diskette, a program that was previously programed by you is loaded in, and your ready to go. With the printer program disk, you program it by answering a series of questions such as giving the CHR# number for your printer for underlining, types of fonts, etc. You can also change the default values for the Atari and other printers for such things as the top margin, line spacing, etc. Very easy and very handy way to configure LP to work with any printer. And you don't have to wait for the DOS to load in!!

The new LP ROM also has added several useful commands to the last version. You can add in the editor special print characters for the control of your printer by using CTRL-V, then a letter, number or CHR# command. You are now able to easily scroll backwards one page at a time. Global replace with or without prompting is now allowed. The handiest new command allows you to chain files together, with the new file starting exactly where the old stops, maintaining all the format information of the first file.

The ROM comes very attractively packaged, well documented, and includes MailMerge, a \$30 utility that allows you to merge a mailing list, change the file from LP to DOS files and back again, as well as other handy utilities. This enables you to use any file with LP; however, the new LP appears to be incompatible with disk-version LP's!. It will soon be available in an 80 column format compatible with the BIT-3 80 column board reported last month (BIT-3, 8120 Penn Ave. So-Suite 548, Minneapolis, Minn. 55431 (612)881-6955). L.J.K. will have a special package price for the two.

DataPerfect, the new database "sister" to LP should be ready soon and mates with LP for form letters, etc.

Is LP worth all that money? If you want a powerful, very easy to use wordprocessor that does just about everything, comes on a ROM for instant use, can be configured easily for any parallel printer, and is compatible with a database, I would say so. LP is particularly good for long documents such as books because of the ability to page and chain chapters, as well as to screen format by page. I have had as many as 25 pages in memory at one time with no problem or slowing of the program. With the Chain function, you could write a book as long as you would like. When the 80 column version comes out, with the BIT-3 board, there will be no competition!

--M. Dunn



ELCOMP's Winfried Hofacker

FILE-IT 2, A Review

(Swift Software, PO Box 641, Melville, NY 11747; \$50.
Written by Jerry White)

Review by Stacy A. Goff

FILE-IT 2 is a collection of programs intended to allow basic record-keeping on your Atari Computer. It requires an Atari with a minimum of 1 disk drive, and 32k of memory. A printer is recommended to gain full value from the system, although once-a month access to a friend's printer can be workable.

With this 'database system' there are three record-types you may use; all are simple formatting of 80-column card-images (shades of the old days!):

- * Free-form entry (you must keep track of the positions for each field)
- * Financial formatting for financial transactions
- * Mailing list formatting

As well, one program on the disk allows you to print disk index inserts.

FILE-IT 2 uses an in-memory record-keeping method (as opposed to a note/point keyed random access method) to store and save all data. On a 40k Atari system this allows up to 304 records in one file. On a 32K system one can have 202 records. The advantage of this in-memory system is that file manipulations, such as sorting, record modification, scanning, listing and deletion are more efficiently done. The disadvantages include file size limitations, and the length of time it takes to read and write the entire file at session beginning and end.

On to specifics about the three subsystems.

FREE-FORM ENTRY -- This is the original FILE-IT program, and allows you a free-form 80-column record which you format and fill in however you wish. This approach has unlimited flexibility, but the record formatting must be consistent if the information is to be listed with any meaning. A nice feature would allow the first record to be a data mask, which defines the format of all following records. This data mask could be printed on the screen any time records are added, displayed or changed. The file manipulation capabilities are reasonably good, and may be used with the formatted files of the other two record types. These capabilities include sort, print, select records, list (3 records at a time), fix or delete records. The sort is painfully slow; the system could use a scan feature (display or select all records containing xxx, or xxx in yy positions).

FINANCIAL RECORDS -- Here is the best part of the system, from my point of view. With prompting, you enter date, accounting code, income and expense figures, and two transaction descriptions. You may then produce a monthly Revenue/Expense report, as well as year-to-date summaries. Finally, you may bar chart your monthly balances, on the screen and optionally, the printer. All records may be manipulated with the capabilities explained above, so transactions may be sorted, etc. Some validity-checking is done, although I managed to put alpha and special characters into positions in the value fields. The program does not discern between 12/81 and 12/82, so you must segregate your files by calendar year in order to have meaningful reports. The reports are extremely well done.

MAILING LIST -- Using a specially modified version of the entry and listing programs to allow formatted mailing list entries, this is a handy addition to the package, although user modification is not possible (for instance, addition of a comments field). Again, all file manipulations of the free-form entry program are allowed.

OVERALL COMMENTS: In addition to the observations above, some general comments are in order:

- * I would not consider this a true database management system (of course I have not seen a true DMS yet for any microcomputer). Instead, I would consider this to be a good card-image pigeon-hole system, with flexible data manipulation features.
- * The documentation is good; a schematic showing how the various parts of the program interact would be helpful. Appendix I, page 3 was blank in my copy of the documentation.
- * The programs support the use of Axlon's new Ramdisk, an advantage if you are doing a lot of updating of the same file.

* Error trapping is a little inadequate or confusing at times; especially when dealing with maximum file capacity--there should be a 'memory full' message.

* The system needs an option for a quick index to data files on disk each time you are asked for a filename.

* A start-up package, which allows the programs to be configured to a one-drive or multiple drive system, would eliminate the painful repeated 'drive#' prompt.

* I might prefer to chart my income or expense in the bargraph, rather than my net. Negative figures should be plotted, rather than ignored.

* The human engineering is fairly good, but could be cleaned up, especially in the multiple switching of disks back and forth. I tried putting programs and data on the same disk, and this cleaned things up considerably, but leaves little room for data.

* The financial system and mailing label systems can handle more than the 304 record maximum, however if you wish to perform any file manipulations using the sort, list, fix etc. features, you must use the select feature to build a smaller working file.

CONCLUSIONS: If you are a beginning user, or are now moving from games to practical applications, FILE-IT 2 is a useful, although unsophisticated addition to your program library. For its \$50 pricetag, you receive a disk full of useful programs. Much more sophisticated programs are available, but they cost twice as much, and are not tailored to such specific applications as is FILE-IT 2.

**MicroCue
Printer Spooler**
POB 7624, Atlanta, GA 30357
(404)874-8366

I recently picked up a Printer spooler for my Atari/Centronics 739 printer, but unfortunately the instruction book had been lost by the dealer. I called the company, who was extremely helpful; and when I hooked it up wrong, they could tell me on the phone which wire was shorted out. A printer spooler allows you to send data to a buffer at computer speeds, then start printing while you get your computer back for other purposes. It's almost like having two printers, it speeds everything up so much. They have several models from the basic 16K one port to 256K 4 port. The model I have will allow input serial/parallel, and output serial/parallel, so you can use a word-processor designed for parallel output with a serial printer. You can also program it with the new model (not mine) to convert characters, so any word-processor could be used for any printer!! Prices range from \$299 to \$899. Well worth looking into, especially if you want to use a slow letter-quality printer with any of the available Atari Word-processors.

M. Dunn

INSTEDIT

"Instedit" is an excellent character generator program written in Basic and machine language. Sheldon Leemon is the author, and he won 2d place in Atari's 1981 competition with this one. It's available from APX, Box 427, 155 Moffett Park Dr., Sunnyvale, CA 94086 for \$17.95. It requires a joystick.

A 21-page user's manual is provided which fully explains all the powerful functions available when using this program.

The screen displays 6 colors and an 8x8 grid in which to design or modify characters. One of 15 different commands may be executed upon the pushing of a key. One may (E)dit, (S)ave, (L)oad, (W)rite an entire character set. One may also operate upon a single character: (A)tari returns the Atari character set; (B)lank out the character; (C)opy any other character onto the one being modified; (I)nvirt it; (M)irror it; (R)estore the character originally chosen to edit; or, (T)wist the character through 90 degree turns.

The screen prompts you with "Which character to edit?" (or copy) when (E) or (C) is pushed. The 4 cursor arrows allow you to move the character around on the 8x8 grid. The joystick moves the cursor on the grid, and the trigger places or removes a block of color. When a letter is chosen to edit, the screen also keeps an example of the original character, and its ASCII code number, in the upper right hand corner.

The lower half of the screen keeps a continuous display of all current characters in the set. A 4-line window at the bottom also displays the current character being created in 5 different modes simultaneously. The <OPTION> key lets you enter and exit a "memo pad" mode which you can use to print strings of characters and build larger characters by combining them. This mode supports character modes not even available in Basic.

This program performs for me a very powerful utility function. I've not been able to discover any bugs or problems trying to get the program to do what the manual says it can do.

-- Jim Bumpas

SANTA CRUZ SOFTWARE'S PLAYER MISSILE GRAPHICS TUTORIAL

REVIEW
by
Chuck Ross

This Tricky Tutorial from Santa Cruz Educational Software is number 5 in their series of tutorials on programming with your Atari.

The program begins very at a very basic level and proceeds to explain to the uninitiated about player-missile graphics. The programs are self-prompting and effective as tutorials. All together there are 14 programs on DISK or TAPE and about 50 pages of documentation including complete listings of all the programs so you can scan the code easily while learning about player missile graphics. Some of the areas covered are memory area of PM graphics, how to draw players, using missiles as players, movement of players (including animation), size selection, resolution (single/double), priority and collision registers.

Included in the package is a game using some of the techniques you will learn in the tutorial. Also a PM graphics design program, which displays all five possible players and moves them for multiple-color-player. All in all a very good package (especially for those who have gotten lost in the numerous PM magazine articles).

So long for now,

Your ACE exchange Librarian
Chuck Ross

*** MINI-REVIEWS***

by Steve Berg
Philippine Islands

RASTER BLASTER BUDGE CO. \$35 32K DISC

Well you pinball wizards can now save your quarters and gas and play at home with your ATARI and Raster Blaster. This budge co. game is identical to the apple version you've seen advertised. It will allow one to four players and you control the flippers using two joystick fire buttons. It has an easy and hard version and best of all the action is suprisingly pinball-like. It has nice but not super sound effects. I recommend it for all of you who can not pass up that arcade without leaving a few quarters behind.

CROSSFIRE

Online Systems
32k Disc \$30

Crossfire is an excellent one player chasem/shootem game where you are being chased around horizontal and vertical "streets". Using the joysticks (or keyboard) you can move and shoot in four directions. Each of the approximately 12 chasers must be shot four times before it is destroyed. Each time you clear the screen you get a new batch and fewer bullets before you must reload. It is a great arcade game at which sooner(probably) or later you'll be wiped out.

SAVMOV REVISITED

by

Ed Chastain

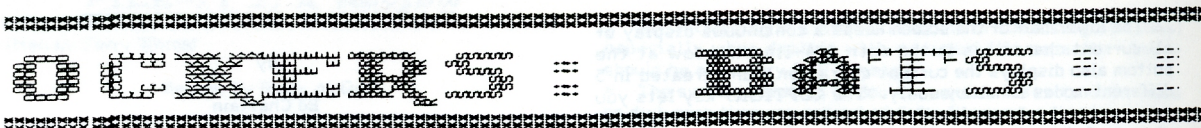
This article further discusses the use of SAVMOV from the January 1982 issue of the ACE Newsletter. Several people have had difficulty understanding what starting address to use with their disassemblers. When I wrote the January article, I assumed the people interested in using it had knowledge of the structure of cartridge programs. Unfortunately, this is not always the case. Prompted by a phone call from Randy in Florida, I am now providing additional background on cartridge programs. Those who have the Operating System reference manual should look at Chapter 10 as a reference for this article. There are six bytes in a cartridge program which provide additional information about the program for the Operating System. These addresses are \$BFFA through \$BFFF in the cartridge. After running through the SAVMOV process they correspond to addresses \$5FFA through \$5FFF. Addresses \$BFFA and \$BFFB, contain the value of the low and high bytes of the cartridge's starting address. This value is also the starting address where you should run your disassembler. Address \$BFFC always contains a zero; this is how the Operating System verifies there is a cartridge present. Address \$BFFD is the option byte, used by the Operating System to determine if it should boot DOS or not. Addresses \$BFFE and \$BFFF contain the value of the low and high bytes of the cartridge's initialization address. This value usually points to an RTS (assembler instruction). If there is an initialization routine then you should run your disassembler at the value stored there. Please remember \$A--- will be at \$4---, and \$B--- will correspond to \$5---.

When running your disassembler, all short branch instructions (BMI, BNE, BEQ, etc.) are based on the relative displacement (+ or - so many bytes from where you are at). They will be shown by your disassembler as addresses between \$4000 and \$5FFF. Remember only the relative displacement is important, not the actual address shown.

The JMP and JSR assembler instructions generally use actual addresses, i.e. \$xxxx. This address will not be changed by the disassembler. For example, if your disassembler says JMP \$BC1D, this means JMP \$5C1D in the relocated program. The value \$BC1D will not be changed since it is an absolute address and not a relative displacement.

Scattered among the cartridge addresses will be text or data which are used by the cartridge program for error messages, character sets, etc. When your disassembler runs through one of these areas, since it is data and not machine instructions, it will appear as meaningless garbage. If your disassembler has a text or ASCII mode, then try it, you may then better understand what that section is.

```
1 REM A STRANGE PROGRAM BY A STRANGE PROGRAMMER
PAGE 1
2 REM BY JERRY WHITE
10 GOTO 30
20 POSITION 0,0:FOR ME=1 TO 99:
CHR$(INT(RND(0)*155)):NEXT ME:POKE 766,0:RETURN
30 GRAPHICS 0:POKE 752,1:POKE 710,0:POKE 764,255:POKE
82,5:
40 ? : ? "RANDOM INSANITY BY JERRY WHITE"
50 ? : ? " PRESS ANY KEY TO BEGIN"
60 ? : ? " THEN PRESS ANY KEY TO STOP"
70 IF PEEK(764)=255 THEN 70
80 X=PEEK(106)-1:Y=PEEK(756):POKE 764,255
90 GRAPHICS 0:POKE 82,0:POKE 83,39:POKE 766,1:POSITION
0,0:GOSUB 20
100 POKE 756,INT(RND(0)*X):SETCOLOR 2,INT(RND(0)*16),0
110 P=INT(RND(0)*240)+5:FOR V=8 TO 0 STEP -.2:POKE
709,INT(V*2)
120 SOUND 0,P,10,V:SOUND 1,P+2,10,V:SOUND
2,P+4,10,V:SOUND 3,P+6,10,V:NEXT V
130 IF PEEK(764)=255 THEN 100
140 POKE 756,Y:POKE 764,255:GRAPHICS 0:POKE 82,2: ?
"STRANGE, WASN'T IT?" :END
```

**** BATS ****

by
Stan Ockers, Lockport, IL

In the March '82 issue of 'Softside' you'll find an excellent program called 'Outer Space Attack' by Sheldon Leemon. This program uses a number of specialized techniques including an interesting way of handling Player Missile Graphics. The method used is to locate the player missile area where a set of strings are stored in memory. Using string operations one can then insert players, move them vertically, or erase them quite easily. About the only disadvantage of this method is it's not hooked into the vertical blank interrupt so movement is not as 'flicker free' as it might be.

I've used this method in a program I call 'Bats'. Line 140 sets up the PM area; the string F\$ is used to make sure the player missile area falls on a 1K boundary (double line resolution). Notice how easy it is to clear an area of memory as in line 170. The missiles are used as insects and a very simple vertical blank routine is used to update their horizontal positions every 1/60th of a second (lines 179-190). The cavern is made by redefining a few characters in the character set and using these to print in GR. 2 mode.

This program handles a problem which has bothered me for some time. If you once initialize PM graphics and then do a Graphics command (GR. 2), you end up with the player missiles being vertical stripes which run the full length of the screen. Reversing the PM initialization doesn't seem to help. The problem is the chip which feeds the screen. It has some registers which are normally being filled by ANTIC during PM graphics. This chip can be disconnected from ANTIC (0 in 53277) but the registers still have information in them which results in the vertical lines. These registers (53261-53264) have to be cleared with pokes as in line 1000. PM graphics is then reinitialized after the graphics mode change (line 2000)

You may have noticed a lack of attention to color in some of my programs. There are two reasons for this: (1) Colors vary from set to set and the ones I choose might not look quite right on your set; and, (2) I normally use a black and white set and am more interested that things show up well in black and white. I have just used the default colors in 'Bats' and there is plenty of room for improvement.

That's it!

-Stan

```
1 REM *****
3 REM * **
4 REM * ACE NEWSLETTER **
5 REM * MAR 1961 **
6 REM * 3662 VINE MAPLE DR **
7 REM * EUGENE, OR 97405 **
8 REM * $10 YEAR **
9 REM *****
20 REM *****
30 REM * BATS **
40 REM * S.O. 3-82 **
50 REM *****
100 DIM Z$(32):FOR I=1 TO 32:
  READ A:Z$(I)=CHR$(A):NEXT I:G
  OSUB 32000:CLR
102 DATA 104,104,133,204,104,1
  33,203,104,133,206,104,133,205
  ,162,4,160,0
104 DATA 177,203,145,205,136,2
  08,249,230,204,230,206,202,208
  ,240,96
110 GOSUB 3000
130 TRAP 1301? "# PLAYERS ""#P
  OKE 764,255:INPUT NP
138 REM ** PM GRAPHICS **
139 REM ** SEE MAR '82 SOFTSID
  E P. 69 **
140 DIM D$(1),F$(INT(ADR(D$)/
  1024)+1)*1024-ADR(D$)-1),PM$(3
  84),M$(128),P$(128),MM$(8)
150 RESTORE 160:FOR I=1 TO 8:R
  EAD A:MM$(I)=CHR$(A):NEXT I
160 DATA 3,3,12,12,48,48,192,1
  92
170 PM$=CHR$(0):PM$(384)=CHR$(
  0):PM$(2)=PM$:M$=PM$:P$=M$
172 REM ** MISSILE COLORS **
174 POKE 704,14:POKE 705,39:PO
  KE 706,54:POKE 707,70
179 REM ** VBI ROUT. TO MOVE M
  ISSILES **
180 FOR I=1536 TO 1566:READ A:
  POKE I,A:NEXT I
190 DATA 104,160,14,162,6,169,
  7,76,92,228,90,120,150,180,162
  ,3,222,10,6,189,10,6,157,4,208
  ,202,16,244,76,98,228
199 REM ** BAT IMAGES **
200 DIM BATDN$(5):BATDN$=P$:FO
  R I=2 TO 4:READ A:BATDN$(I,I)=
  CHR$(A):NEXT I
210 DATA 24,165,66
```

```
220 DIM BATUP$(5):BATUP$=P$:FO
  R I=2 TO 4:READ A:BATUP$(I,I)=
  CHR$(A):NEXT I
230 DATA 66,165,24
250 POKE 54279,ADR(PM$)/256:PO
  KE 559,46:POKE 53277,3:POKE 62
  3,4:A=USR(1536)
259 REM ** STALACTITES AND STA
  LAGMITES **
260 DIM C$(42),U$(42):C$="****
  %$*****&"U$=")(*****"
  "FOR I=1 TO 14:C$(I+14)=CHR$(
  ASC(C$(I))-32)
270 U$(I+14)=CHR$(ASC(U$(I))-3
  2):C$(I+28)=CHR$(ASC(C$(I))+12
  8):U$(I+28)=CHR$(ASC(U$(I))+12
  8):NEXT I
272 DIM P(NP),SCORE(NP),TOTAL(
  NP),BN(NP),BONUS(NP)
275 W=7:P=0:POKE 82,0
278 FOR I=1 TO NP:SCORE(I)=0:T
  OTAL(I)=0:BN(I)=3:BONUS(I)=100
  0:NEXT I:NXTCV=300*NP
279 REM ** CHANGE WIDTH OF CAV
  ERN **
280 IF W>3 THEN W=W-1
282 GOSUB 480
284 P=P+1:IF P>NP THEN P=1
286 IF BN(P)=0 THEN 284
287 M$=PM$:FOR I=0 TO 3:M$(YST
  +5*W+W*(3-I))=MM$(2*I+1,2*I+2)
  :NEXT I
289 REM ** MAIN LOOP **
290 POKE 656,1:POKE 657,22:"
  Pull Joystick"
295 IF STICK(0)>13 THEN 295
299 REM ** SCORECARD **
300 ? CHR$(125):GOSUB 700:POKE
  656,0:POKE 657,26:"PLAYER #
  "P
301 POKE 656,1:POKE 657,24:"
  Round Total":GOSUB 860
306 POKE 53248,30:YPOS=YST+20:
  POKE 53278,0:T=0:DIS=12
310 FOR XPOS=47 TO 200:POKE 53
  248,XPOS:IF STRIG(0)=0 THEN YP
  OS=YPOS-1:P$(YPOS)=BATUP$
320 IF STRIG(0)=1 THEN YPOS=YP
  OS+1:P$(YPOS)=BATDN$
330 IF PEEK(53256)>0 THEN POKE
  1546,0:TOTAL(P)=TOTAL(P)-100:
  GOTO 420
```

```
340 IF PEEK(53257)>0 THEN POKE
  1547,0:GOSUB 900
350 IF PEEK(53258)>0 THEN POKE
  1548,0:GOSUB 900
360 IF PEEK(53259)>0 THEN POKE
  1549,0:GOSUB 900
370 IF PEEK(53252)>0 THEN 420
390 NEXT XPOS:P$=PM$
395 IF SCORE(P)<300 THEN 306
400 TOTAL(P)=TOTAL(P)+SCORE(P)
  :SCORE(P)=0:GOSUB 870
402 IF TOTAL(P)>BONUS(P) AND B
  N(P)<4 THEN BONUS(P)=BONUS(P)+
  1000:BN(P)=BN(P)+1:GOSUB 700:D
  IS=10:T=30:GOSUB 630
403 FOR I=1 TO 30:GOSUB 880:FO
  R J=1 TO 30:NEXT J:GOSUB 860:N
  EXT I
404 IF P=NP THEN 280
410 GOTO 284
419 REM ** LOSE A BAT **
420 DIS=10:T=9:GOSUB 630
430 YPOS=YPOS+1:P$(YPOS)=BATDN
  $:POKE 53278,0:SOUND 1,YPOS,10
  ,10:IF PEEK(53252)=0 THEN 430
445 GOSUB 640:P$=PM$:SCORE(P)=
  0:BN(P)=BN(P)-1:GOSUB 700:IF B
  N(P)=0 THEN POKE 656,0:POKE 65
  7,6:" "GOSUB 660
460 GOSUB 850:FOR I=1 TO NP:IF
  BN(I)>0 THEN 403
470 NEXT I:GOSUB 1000:GRAPHICS
  17:POSITION 5,2: ? #6:"Game Ov
  er":FOR I=1 TO NP:POSITION 3,2
  +2*I: ? #6:"Player #":I
472 ? #6:"=":TOTAL(I):NEXT I
  :POSITION 3,23: ? #6:"PRESS ANY
  KEY"
475 FOR I=1 TO 300:NEXT I:GOSU
  B 655:IF FL=0 THEN 475
476 GOTO 275
479 REM ** DRAW CAVERN **
480 GOSUB 1000:GRAPHICS 2:GOSU
  B 2000:POKE 77,0
490 DL=INT(RND(0)*(8-W))+1:YST
  =8*(DL+1)
500 FOR X=0 TO 19:GOSUB 590:Y=
  0:FOR I=R+7-DL TO R+6:POSITION
  X,Y: ? #6:C$(I,I):Y=Y+1:NEXT I
  510 FOR I=1 TO W:POSITION X,Y:
  ? #6:" "Y=Y+1:NEXT I
  520 IF DL+W>=10 THEN Y=Y-1:POS
  ITION X,Y: ? #6:"":GOTO 540
```



```

530 GOSUB 590:FOR I=R TO R+9-D
L-W:POSITION X,Y:Y? #6;U$(I,D):
Y=Y+1:NEXT I
540 IF DL<=1 THEN DL=2:GOTO 57
0
550 IF DL=10-W THEN DL=9-W:GO
TO 570
560 DL=DL+INT(RND(0)*3)-1
570 NEXT X
580 RETURN
590 R=INT(RND(0)*6)*7+1:RETURN

629 REM ** SOUND SUBR'S **
630 FOR I=15 TO 0 STEP -1:SOUN
D 0,I,DIS,I:FOR J=1 TO T:NEXT
J:NEXT I:RETURN
640 FOR I=10 TO 2 STEP -2:SOUN
D 0,RND(0)*255,3,I:SOUND 1,RND
(0)*255,8,I:FOR J=1 TO 30:NEXT
J:NEXT I
650 SOUND 0,0,0,0:SOUND 1,0,0,
0:RETURN
655 RESTORE 697:LS=30:LL=5:GOS
UB 664:RETURN
660 RESTORE 690:LS=20:LL=10
664 FL=0
665 READ I,J:IF I=3 THEN RETUR
N
666 IF I=0 THEN 670
667 IF PEEK(53775)<255 THEN FL
=1:RETURN
669 SOUND 0,I,10,10:SOUND 1,I-
2,10,6
670 FOR I=1 TO J:FOR K=1 TO LS
:NEXT K:NEXT I:SOUND 0,0,0,0:S
OUND 1,0,0,0
675 FOR I=1 TO LL:NEXT I:GOTO
665
680 RESTORE 692:LS=12:LL=12:GO
SUB 664
681 IF FL=1 THEN RETURN
682 RESTORE 694:GOSUB 664
683 IF FL=1 THEN RETURN
684 RESTORE 692:GOSUB 664
685 IF FL=1 THEN RETURN
686 RESTORE 696:GOSUB 664
687 IF FL=1 THEN RETURN
688 FOR I=1 TO 300:NEXT I:GOTO
680
690 DATA 243,4,243,4,243,1,243
,4,204,4,217,1,217,4,243,1,243
,4,255,1,243,6,3,3
692 DATA 243,1,217,1,204,1,182
,1,162,1,204,1,162,1,0,1,173,1
,217,1,173,1,0,1,182,1,230,1,1
82,1,0,1
693 DATA 243,1,217,1,204,1,182
,1,162,1,204,1,162,1,121,1,3,3

694 DATA 136,1,162,1,204,1,162
,1,136,4,3,3
696 DATA 162,1,204,1,162,1,121
,1,243,4,3,3
697 DATA 81,4,85,2,102,1,108,1
,121,6,108,1,102,1,81,2,81,2,8
5,2,102,1,108,1,121,8
698 DATA 108,2,91,2,102,2,108,
2,121,1,128,1,121,1,108,1,102,
2,121,2,81,4,102,4,121,8,3,3
699 REM ** SUBR. TO INDICATE B
ATS LEFT **
700 POKE 656,0:POKE 657,6: "
" :POKE 657,6:FOR I=1 TO
BN(P): " + " :NEXT I:RETURN
850 POKE 656,1:POKE 657,5: SC
ORE(P): " :RETURN
860 POKE 656,1:POKE 657,12: T
OTAL(P): " :RETURN
870 POKE 656,1:POKE 657,5: "
" :RETURN
880 POKE 656,1:POKE 657,12: "
" :RETURN
900 GOSUB 630:POKE 53278,0:SCO
RE(P)=SCORE(P)+25:GOTO 850
999 REM ** SUBR. TO REMOVE PM
GR. **
1000 POKE 53277,0:POKE 54272,0
:FOR I=53261 TO 53264:POKE I,0
:NEXT I:RETURN
1999 REM ** SUBR. TO INSERT PM
GR. **
2000 POKE 53277,3:POKE 559,46:
START=(PEEK(106)+1):POKE 756,5
TART
2009 REM ** ALTER DISPLAY LIST
**
2010 A=PEEK(560)+256*PEEK(561)
2020 IF PEEK(A)<66 THEN A=A+1
:GOTO 2020
2030 POKE A,70:POKE A+3,6:POKE
A+4,6:POKE A+5,6
2040 RETURN
3000 ? :? :? CHR$(125):? " + +
+ + + + + BATS + + + + + + +
"?
3010 ? " The object is to fl
y your bat":? "through a caver
n eating flying"
3020 ? "Insects along the way.
The only":? "control you hav
e is the FIRE":? "button. Pus
h it to go up."
3030 ? "Release it and your ba
t starts":? "to fall." :? : "
A round lasts 300 points or"

3035 ? "until you lose a bat.
Each time":? "you hit a stila
ctite or a"
3040 ? "stilagmite you lose on
e bat":? "and points earned fo
r that round."
3050 ? "Each insect you eat sc
ores points":? "but do not eat
the bright insect the"
3060 ? "color of your bat.. it
is poison!":? "Eat him and yo
u lose round points":? "plus 1
00 more and a bat!":?
3065 ? "PRESS ANY KEY":GOSUB 6
80: CHR$(125):? :?
3070 ? " After a time the ca
vern will":? "become narrower
and scores for the"
3080 ? "last round will be add
ed to totals.":? "Every 1000 p
oints you are given"
3090 ? "another bat, 4 maximum
. Lose":? "all your bats and
you are out."
4000 ? "The game is over when
everyone is":? "out of bats.
Everyone uses the":? "leftmost
joystick":? :RETURN
31999 REM ** CHANGE CHARACTER
SET **
32000 POKE 106,PEEK(106)-5:GRA
PHICS 0:START=(PEEK(106)+1)*25
6:POKE 756,START/256:POKE 752,
1
32010 ? "INITIALIZING ....."
32020 A=USR(ADR(Z$),57344,STA
RT):RESTORE 32100
32030 READ X:IF X=-1 THEN REST
ORE :RETURN
32040 FOR Y=0 TO 7:READ Z:POKE
X+Y+START,Z:NEXT Y:GOTO 32030
32100 DATA 32,255,255,127,127,
126,62,62,60
32101 DATA 40,60,28,28,24,8,8,
8,8
32102 DATA 48,255,127,126,60,5
6,24,8,8
32103 DATA 56,8,24,28,124,124,
254,254,255
32104 DATA 64,60,126,126,126,1
26,126,127,255
32105 DATA 72,16,16,16,16,16,2
4,60,60
32106 DATA 80,255,255,255,255,
255,255,255,255
32107 DATA 88,0,24,24,165,165,
66,66,0
32108 DATA -1

```



Fred Thorlin of APX

Chicken Dressing

by Hank Hirschfeld, Northvale, NJ

One of the things that is nice about arcade games is the ability to save high scores after the game is turned off. This is something that can be incorporated in many of the games that are on disk. The ability to open and close files on a disk allows us to set aside a portion of our disk for maintaining high score data. Having only owned my computer for a few months, I'm sure that many of you will find ways to improve on this concept. In fact, my purpose is to have more programmers put this feature in future games on disk. I must credit the article in the April 1982 issue of Creative Computing called "Using Disks with Atari BASIC" by Johnson and Humphrey for stimulating my interest.

As an example, I have made a program which can be merged with the very popular Stan Ocker's CHICKEN game. The exact changes will vary from game to game; you will have to adapt the program as required, but that's most of the fun anyway!

I have incorporated a way to keep the high score, generate a sound indicating a new high, and a way to return the high score to 0 using a code.

HHS is the high score on the disk
SCORE is a variable for the game score
HIGH is a variable for the high score
XX is a delay loop for the sound

Line 250—Brings high score on disk into program at the start
Line 719—Compares score with disk high score; if we have set new high we will go into subroutine

Line 731—If you hold down OPTION, SELECT and START at the same time right after "CHICKEN'S DEAD" and hold down until "press fire button" appears, you will reset high score on disk to zero.

Line 920—End of game, returns to 719 to make comparisons of scores.

Line 9000, 9010—Opening of file to obtain high score on disk.

Line 9020—Compare scores

Line 9040, 9050—Write new high score to disk

Line 9060-9080—New high music

Line 9200, 9210—Writes zero score to disk and removes previous score.

You must also establish a file on the disk for keeping this data, the file used in this program is called "D1:CHICKEN.DAT". To do this, type the following with your disk inserted, use direct mode so line numbers are not needed:

OPEN #7,8,0,"D1:CHICKEN.DAT"

then press RETURN and type:

INPUT #7,0;CLOSE #7

then press RETURN.

If you look at your DOA menu, the file should be there.

```
249 REM CHICKEN DRESSING by Hank Hirschfeld
250 GOSUB 9000
719 IF SCORE>HHS THEN GOSUB 9040
731 IF PEEK(53279)=0 THEN GOSUB 9200
920 GOTO 719
9000 OPEN #7,4,0,"D1:CHICKEN.DAT"
9010 INPUT #7,HHS;HIGH=HHS
9020 IF HHS>SCORE THEN GOTO 9100
9030 CLOSE #7
9040 OPEN #7,8,0,"D1:CHICKEN.DAT"
9050 ? #7;SCORE:HHS=SCORE
9060 XX=0
9070 SOUND 3,200-XX,10,8;XX=XX+1;IF XX=200 THEN GOTO 9100
9100
9080 GOTO 9070
9100 CLOSE #7;RETURN
9200 OPEN #7,8,0,"D1:CHICKEN.DAT"
9210 ? #7;0;CLOSE #7;RETURN
```

I leave it up to others to incorporate listing more than one high score and allowing the insertion of the initials of the players with high scores.

PICTURE THIS!

review by Ruth Ellsworth

David D. Thornburg is the author of this cleverly written introduction to computer graphics. As stated by the author, the book is for kids of all ages, and the text is presented with a rare sense of humor that holds the interest of the young and old alike. The illustrations are simple, and the text clear, yet a spirit of excitement and discovery is sustained throughout.

PICTURE THIS! begins with the most basic instructions and takes the reader through the creation of simple pictures to simple games. Several geometric concepts are introduced using the discovery method of teaching, giving us the opportunity to "play detective" which the children found particularly delightful. They also were very excited about the way in which the author taught them how to draw simple pictures and then to move their creations across the screen. I was very impressed with the step by step method of instruction used, and the way that the author taught by example the annotation of programs (see the demonstration PILOT program). It greatly increased the children's ability to write their own programming because they could review their own programs and see what they had done and why.

We encountered only three problems in the entire book which I list here in what we considered their increasing degree of difficulty caused by the problem. There were several printing errors which misplaced pictures in relation to the text. The author failed to teach the kids that a program begun in AUTO could be continued simply by typing AUTO followed by the number with which they wished to restart the program. Repeated use of the system reset key is encouraged which is poor programming technique and quickly becomes a bad habit.

I used this text on my own kids and borrowed sundry friends. It was our conclusion that this book is a definite MUST for anyone owning the Atari PILOT program. I would further suggest that it be used as the first introduction to PILOT for either children or adults, followed by the PILOT student manual which is adaptable enough to be used as a second text.

HARDWARE

The circuit diagram was sent to me from one of our many Australian members. I could not call him to ask questions! If one of you brave souls could make it, let me know of your experiences, etc. Would really appreciate a printed circuit diagram that I could put in the Newsletter for all. Maybe someone could even make a kit and sell to members for a nominal price.

Parts list

Misc.

1 x IC LM324N Quad OP AMP
4 x Silicon Diodes, Low power

Resistors (Ohms)

1 x 820
2 x 1k
3 x 4.7k
1 x 5.6k
1 x 7.5k
1 x 10k
2 x 15k
1 x 56k
1 x 82k
1 x 240k
1 x 270k
2 x 330k

Capacitors (Disk Ceramic)

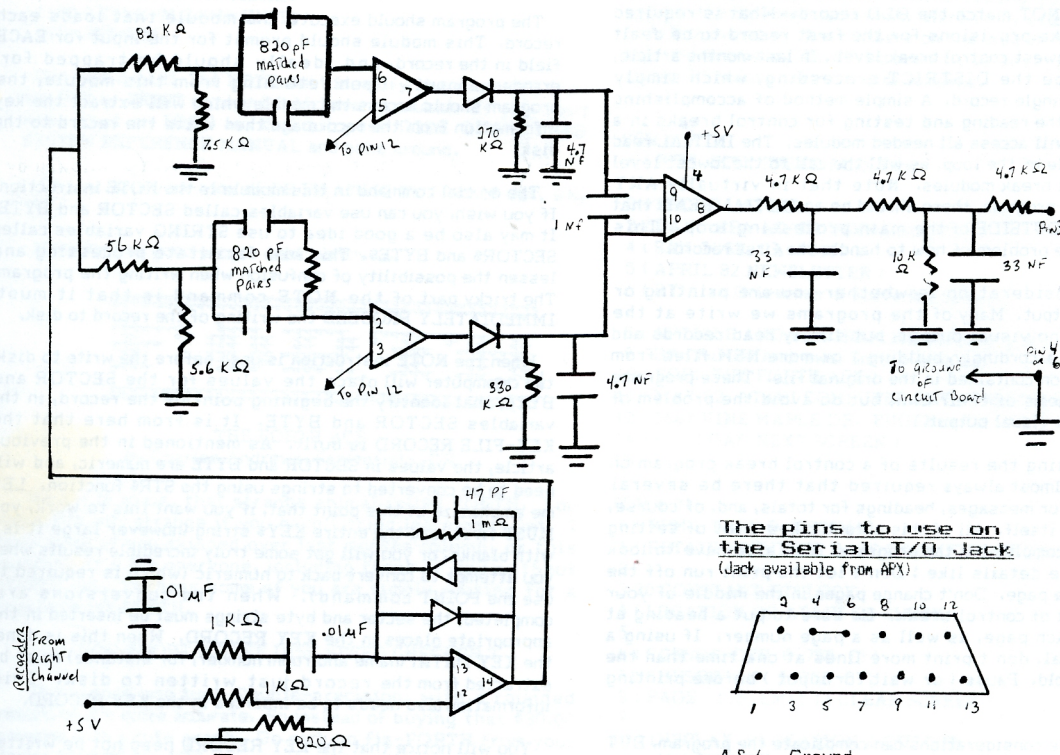
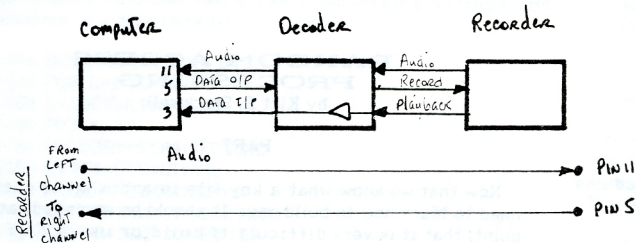
4 x 820pF
1 x 47pF
1 x 1nF
2 x 4.7nF
2 x 33nF
2 x .01uF

Other

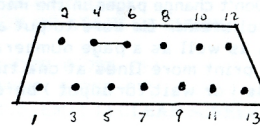
1 x 5volt relay
1 x jiffy box

Cassette Recorder Interface for the Atari 400/800 Computer

by Denis Biddle, Australia



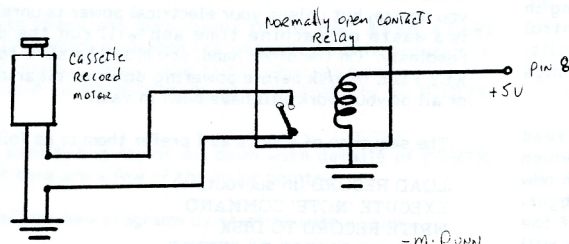
The pins to use on the Serial I/O Jack (Jack available from APX)



4 & 6 Ground

- 8 Motor control of cassette and +5 volts for external
- 5 Data output, connect directly to record in of cassette. Adjust record level for no distortion
- 3 Data input, connect to the decoder
- 11 Audio input, the left channel of your recorder.

Computer Controlled Auto on/off of cassette motor (optional)...



NOTE

Do not connect the ground of the computer to the ground of the recorder

Use shielded wire to interconnect between computer and recorder

There are 5 wires going to the cassette recorder, 2 for motor, 1 for Audio, 1 for right channel record, and 1 for replay

FEAR FROM KIRK

STRUCTURE IN BASIC

By Kirt E. Stockwell

Ahh Yess, the Venerable Control Break, my wife had one just before they took her away.....

As you looked at the mock listings in the last structure article, you probably asked yourself (you didn't ??) what happens when the first record is read in, since the first record read in CAN NOT match the OLD record. What is required here is to make provisions for the first record to be dealt with on the lowest control break level. In last month's article, that would be the DISTRICT processing, which simply processes a single record. A simple method of accomplishing this is to do the reading and testing for control breaks in a loop, which will access all needed modules. The INITIAL read will be outside of the loop, as will the call to the lowest level of the control break modules. Note that in virtually ANY control break program, there should be an INITIAL READ that is performed OUTSIDE of the main processing loop. This simplifies the problem of how to handle the first record.

Another consideration is whether you are printing or displaying output. Many of the programs we write at the college have no visual output, but simply read records and process them accordingly, building 1 or more NEW files from the information contained in the original file. These programs present problems of their own, but do avoid the problem of how to handle visual output.

When printing the results of a control break program on paper, it is almost always required that there be several headings, error messages, headings for totals, and, of course, the data line itself. All of this output takes a lot of setting up, but also complicates the program. You also have to look out for little details like: Don't let the print run off the bottom of the page. Don't change pages in the middle of your smallest unit of control break. Be sure to put a heading at the top of each page, as well as a page number. If using a video terminal, don't print more lines at one time than the screen can hold. Pause (or wait for input) before printing more.

All of these considerations can complicate the program, BUT need not be too troublesome if you remember to write the entire program in modules. Above all else, that is the key consideration in STRUCTURED programming, and is essential in control break programs. Now, look at the mock listing in last month's issue, and use it as a basis for writing a control break program. Then, if you can do that, start adding frills, like headings, fancy output and whatever. If you keep things structured properly, you should have no trouble.

What happens, though, when you run out of records to read ?? Well, on the ATARI, you get an ERROR condition, which normally would blow the program up. But we need to learn how to deal with that, as the last record is an important trigger, as it is used to break out of the work loop. Once out of the loop, we can access the FINAL-TOTALS routine, which will then process the last record stored in the OLD RECORD area. We can then drop through to the close-out routine, which will close the files and take care of any minor details.

Remember that the INITIAL READ and PROCESS, the call to the FINAL-TOTALS, and the call to the CLOSE-OUT ROUTINE should be in the MAIN-BODY. Everything else should be in modules. As a rule, the smaller the better, but don't get crazy.

Ponder on these things, my children and come back next month. Bye !

RANDOM ACCESS PROCESSING

by Kirt E. Stockwell

PART 2

Now that we know what a key file is and what it does, we need to learn how to build one. It should be stressed, at this point, that it is very difficult to build or use a KEY file without using modular programming, which we have discussed in the STRUCTURE articles.

The program should execute the module that loads each record. This module should prompt for the input for EACH field in the record, and ideally should be trapped for erroneous input. Upon returning from this module, the program should access the module which will extract the key information from the record and then write the record to the disk.

The crucial command in this module is the NOTE instruction. If you wish, you can use variables called SECTOR and BYTE. It may also be a good idea to use STRING variables called SECTOR\$ and BYTE\$. This will facilitate processing and lessen the possibility of confusion when writing the program. The tricky part of the NOTE command is that it must IMMEDIATELY PROCEED the writing of the record to disk.

When the NOTE instruction is used before the write to disk, the computer will place the values for the SECTOR and BYTE, that identify the beginning point of the record, in the variables SECTOR and BYTE. It is from here that the KEY-FILE RECORD is built. As mentioned in the previous article, the values in SECTOR and BYTE are numeric, and will need to be converted to strings using the STR\$ function. LET me emphasize at this point that, if you want this to work, you MUST first load the entire KEY\$ string (however large it is) with blanks, or you will get some truly incredible results when you attempt to convert back to numeric (which is required to use the POINT command). When the conversions are completed, the sector and byte strings must be inserted in the appropriate places in the KEY RECORD. When this is done, the KEY DATA, (name and room number, for instance) must be extracted from the record just written to disk. This information also needs to be inserted in the KEY RECORD.

You will notice that the KEY RECORD need not be written to disk immediately, as it is in the form of one LARGE string, and there is room in memory for it in its entirety. You can write each record of the KEY FILE to disk as it is loaded if you desire, but unless your electrical power is unreliable, this is a waste of machine time and will run the disk drive needlessly. On the other hand, you MUST be sure to write the KEY FILE to disk before powering down or clearing memory, or all of your work will have been in vain.

The sequence of events as I prefer them is as follows:

```
LOAD RECORD (in subroutine)
EXECUTE 'NOTE' COMMAND
WRITE RECORD TO DISK
CONVERT SECTOR TO STRING
CONVERT BYTE TO STRING
STORE SECTOR, BYTE IN KEY$
EXTRACT KEYS FROM DATA
(such as NAME and UNIT)
STORE KEYS IN KEY$
LOOP BACK TO LOAD NEXT RECORD
```

Of course, this is just a basic outline, and will have to be worked into the framework of the program you want to write. Some of the program considerations you will have to examine are: Will all data be entered at one time ? Will KEYS or RECORD NUMBER be used to UPDATE or EDIT a record ? Is anyone going to use the program or the files created by it other than yourself ?

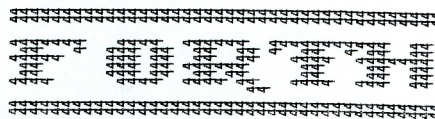
Let's look at the NOTE command as it would be used in the program. In this example, I use the PUT and a loop to write the record to the disk, because the ATARI DOS limits string length in DISK I/O to a smaller figure than the size (300 bytes) of the strings that are my records. (also, it eliminates the complications that can arise from mixing strings and variables in a disk record.)

```
1000 NOTE#1, SECTOR, BYTE
1010 FOR L=1 TO 300
1020 X=ASC(DATA$(L,L))
1030 PUT #1,X;
1040 SECTOR#=STR$(SECTOR)
1050 BYTE#=STR$(BYTE)
1060 Z=RECORDNUMBER*30
1070 KEY$(Z+25,Z+27)=SECTOR#
1080 KEY$(Z+28,Z+30)=BYTE#
1090 NAME$=DATA$(1,20)
1100 UNIT$=DATA$(21,24)
1110 KEY$(Z+1,Z+20)=NAME$
1120 KEY$(Z+21,Z+24)=UNIT$
1130 RETURN
```

One last detail: The NOTE must identify the same IOCB as the PUT or PRINT command (whichever you use). If you don't know what an IOCB is, dig out your DISK OPERATING SYSTEM REFERENCE MANUAL and poke around.

Do you have your copy of DE RE ATARI YET ??? If not, why ???

More for you good people next month.....



THE FORTH ACE
by Charles Andrews, Springfield, OR

Here it is again, that strange animal called FORTH. You know, it's only strange if you haven't tried learning it yet. I hope none of you skipped over last month's column, or this month's. It's for everyone, including those of you without FORTH yet. Perhaps some of you are simply waiting for a reason to get FORTH.

HOW ABOUT A FREE DISK SPEED CHECKER ?

It's a good one too! Double-precision, with a rounded result, that's quite accurate. So instead of buying that \$30.00 program with a cute graph. Go pick up fig-FORTH from your dealer (\$39.95 for APX program). It's still incredible to me to get another complete language and operating system, with so much power, for such a bargain.

I want to thank and give credit to the BAY AREA ATARI USER GROUP, which published the essence of this program in their April issue. I only made slight changes to improve the accuracy and to demonstrate the readability of FORTH programs.

I don't want to bog any of you down with details of FORTH syntax, but here are a few of the major points:

FORTH defines new programs by starting with a colon.

The first word listed after the colon ":" is the name of the new function.

There must be a space separating all words. (Even the : is a word)

All words that follow in the definition are now performed by the single function name.

The definition is finished with a semi-colon.

Example
: name built from other words ;

By choosing descriptive names for the functions your program can be very readable.

Comments should be added for program readability and documentation. Comments appear between parentheses. Comments are completely ignored by the computer and DO NOT take any RAM.

Comment

(anything for HUMAN understanding)

This brings us to our program. FORTH comes with a disk screen editor which is most efficient for writing your programs. The source code, which can be very readable, is saved on disk including comments and documentation. Corrections or changes can be easily made. You then LOAD the disk screens. Then the program is completely compiled into memory and available at any time by name.

Example: RPM <Return>

The comments are ignored when compiled and the program can remain in memory taking very little space. This program uses 263 bytes of RAM. To save typing you only need to include the comments you want. Here is my example to you of a readable program. The numbers on the side of the screens are not line numbers. FORTH doesn't use any. They are simply editor ID numbers for the disk screens.

```
SCR # 30
0 (RPM DISK SPEED CHECKER)
1
2 (PROGRAM FROM)
3 (BAY AREA ATARI USER GROUP)
4 (SAN JOSE, CA 95117)
5 (APRIL 82 NEWSLETTER)
6 (BY Alan Griesemer & Steve Bradshaw)
7
8 (ARTICLE AND PROGRAM MODIFICATIONS)
9 (BY Charles Andrews, Springfield, OR)
10 (FOR: THE FORTH ACE)
11 (MAY 82 ATARI COMPUTER ENTHUSIASTS NEWSLETTER)
12 (3662 VINE MAPLE DR. EUGENE, OR 97405)
14 --> (LOAD NEXT SCREEN)
15
```

```
SCR # 31
0 (RPM) BASE @ DECIMAL
1 (RETAIN PREVIOUS BASE & COMPILE FROM DECIMAL)
2
3 20 CONSTANT RTCLOCK
4 (LOCATION OF ATARI OS CLOCK)
5
6 : CURSOR-OFF 0 755 C!;
7 : CURSOR-ON 2 755 C!;
8 : PAGE 125 EMIT ; (CLEAR SCREEN)
9
10 : DISPLAY (LOCATION & OUTPUT)
11 10 (ROWCRS) 84 C!;
12 6 (COLCRS) 85 !;
13 ." RPM = " . (PRINT);
14
15 -->
```

```
SCR # 32
0 (RPM - CONT)
1
2 : COMPUTE (RPM OF 16 DISK READS)
3 57600 0 (DOUBLE-PRECISION NUMERATOR)
4 RTCLOCK C@ (DENOMINATOR)
5 U/ (DOUBLE-PRECISION DIVIDE)
6 >R (REMOVE INTEGER RESULT)
7 100 > (TEST REMAINDER)
8 IF 1 (ROUND UP)
9 ELSE 0 (NO CHANGE)
10 THEN R> (RETURN RESULT)
11 + (SUM ROUNDED RESULT)
12
13 : READ-SECTOR (READ SECTOR 320)
14 PAD 320 1 R/W;
15 -->
```

```
SCR # 33
0 (RPM - CONT)
1 : RPM (THIS IS THE PROGRAM)
2 BASE @ DECIMAL
3 (RETAIN CURRENT BASE & OUTPUT IN DECIMAL)
4 CURSOR-OFF PAGE
5 BEGIN ?TERMINAL 0= WHILE
```



```

6 ( REPEAT AS LONG AS FLAG = 0 )
7 READ-SECTOR ( TO SEEK HEAD )
8 0 RTCLKOCK C! ( RESET TIMER )
9 16 0 DO READ-SECTOR LOOP
10 ( READ DISK SECTOR 16 TIMES )
11 COMPUTE DISPLAY
12 REPEAT ( TEST FLAG THEN REPEAT IF TRUE )
13 CURSOR-ON BASE ! ;
14 ( RESTORE CURSOR & PREVIOUS BASE END )
15 -->

SCR # 34
0 ( EXECUTE RPM & INSTRUCTIONS )
1
2 : DELAY 32000 0 DO LOOP ; ( 4 SECONDS )
3
4 PAGE , " COMPILING COMPLETE " CR CR
5 , " RPM WILL RUN AUTOMATICALLY " CR
6 , " HIT ANY KEY TO STOP PROGRAM " CR
7 , " OK " DELAY
8
9 RPM
10
11 BASE ! FORGET DELAY ;S
12 ( RESTORE THE BASE THAT EXISTED PRIOR TO
COMPILING )
13
14 ( RPM IS NOW IN YOUR VLIST & WILL RUN ANYTIME )
15 ( ENTER RPM <Return> )

```

Call or write anytime -- Enjoy

-- Charles Andrews --
P.O. Box 1613, Eugene, OR 97440
phone (503) 747-9892

-Not as fancy as Tony Dobre's excellent "Tachmaster"
available from Swifty, but it seems to work well-ed.

SEE RUTH GIVE A
DEMONSTRATION OF PILOT
AT OUR NEXT MEETING IN
MAY!!

Atari Computer Enthusiasts

A.C.E. is an independent computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in education our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

**All ourdarticles, reviews and
programs come from you, our
members.**

Our membership is world-wide in scope; membership fees include the A.C.E. Newsletter, published about \$10 a year. Dues are \$10 a year for U.S., and \$20 a year Overseas Airmail.

President-Kirt Stockwell, 1810 Harris #139, Eugene, Or 97403 503-686-2470

Secretary-Charles Andrews, POB 1613, Eugene, Or 974401613 503-747-9892

Librarian-Chuck and Jody Ross, 2222 Ironwood, Eugene, OR 97401 503-343-5545

Editors -Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97505 503-344-6193
-Jim Bumpas, 1405 W. 26th Ave, Eugene, Or 97401 503-484-9925

**Send a SASE to the Ross' for the
new, updated ACE Library List!!**

**Best of ACE-1981, still available,
Disk or Tape, \$8!**

Bulletin Board

(503)343-4352

We are getting more and more calls and are considering buying the equipment for full-time use. It ties up a computer system as well as a phone line, so now must be limited. Remember, if it rings more than once, hang up as it is not on. I will try to keep in on weekdays, except Wednesday from 11PM to 3PM PST and until NOON Sat PST. Also, SUNDAY from Midnight to about 9AM. But remember, if it rings more than once, either I forgot to put in on or we are using the computer! Let me know if a full-time system would interest you, and if you would be willing to pay a nominal fee for expenses and how much.

-Mike Dunn

April Meeting

Always the second Weds of the Month-May 12 at 7:30PM. At the Northwest Natural Gas Building on Goodpasture Road next to K-Mart off Delta Highway by Valley River. A new

ATARI
COMPUTER
ENTHUSIASTS
3662 Vine Maple Dr Eugene OR 97405

**FIRST
CLASS
MAIL**